IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Minglu Zeng
10/14/2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

I collected data of SpaceX from its websites and try to predict the Falcon 9 first stage. And that's important because SpaceX advantage over conventional rockets is that it's recyclable.

- Problems you want to find answers
- 1. Which factors determine the successful rate of landing?
- 2. Which model is used to predict with higher accuracy?

Section 1

Methodology

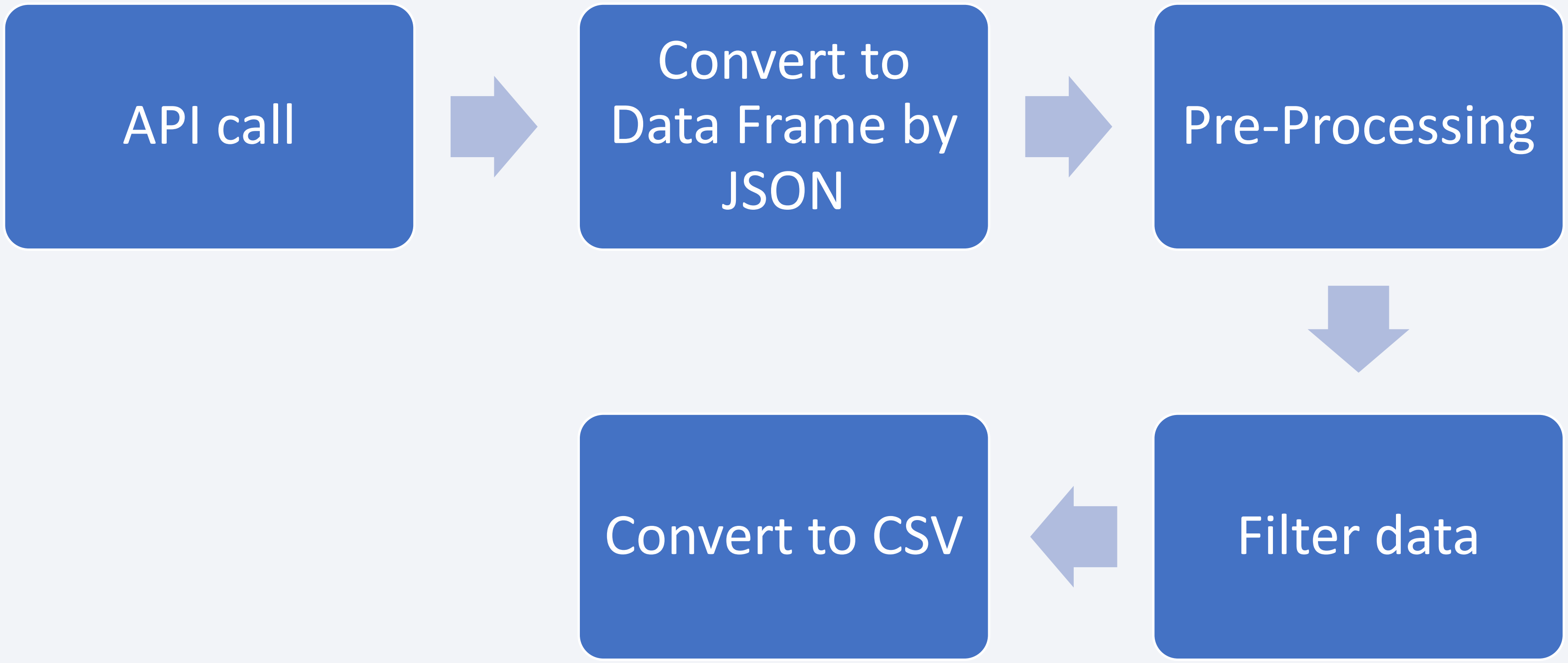
Methodology

Executive Summary

- Data collection methodology:
  - API and Web Scrapping
- Perform data wrangling
  - Data transformation and one hot encoded
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatter plot, line plot and bar plot
- Perform interactive visual analytics using Folium and Plotly Dash
  - Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune, evaluate classification models

Data Collection

Data sets are collected using the API call from SpaceX Websites.

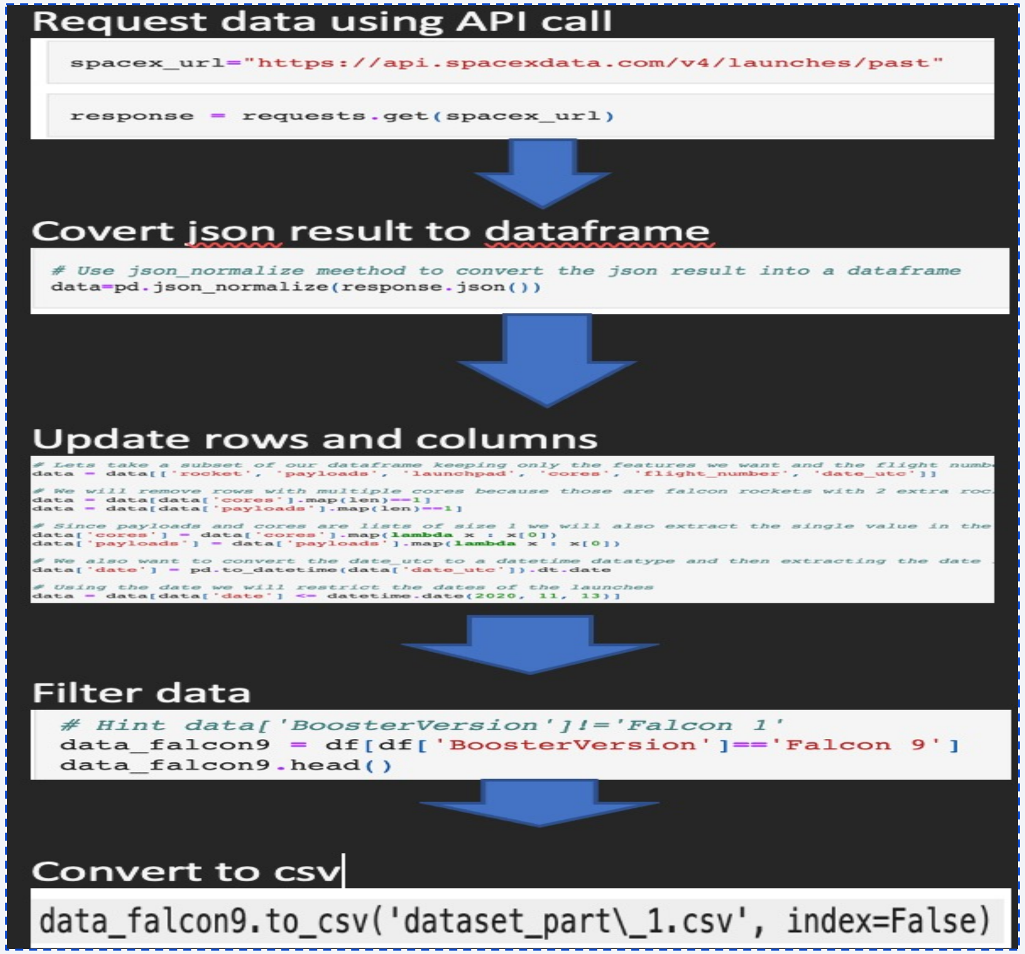


Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

- https://github.com/mingluzeng/spaceX_assignment/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

- https://github.com/mingluzeng/spaceX_assignment/blob/main/jupyter-labs-webscraping.ipynb

1.Get responses from HTML

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

2.Create Beautiful Soup Object

```
# Use BeautifulSoup() to create a BeautifulSoup
soup = BeautifulSoup(response, 'html.parser')
```

3.Finding tables

```
# Use the find_all function in the Beautifu
# Assign the result to a list called `html_t
html_tables = soup.find_all("table")
print(html_tables)
```

4.Getting columns name

```
column_names = []

# Apply find_all() function with `th` element on fir
# Iterate each th element and apply the provided ext
# Append the Non-empty column name (`if name is not
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
        name = extract_column_from_header(temp[x])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

5.Creating dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each va
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6.Appending data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
```

7.Coverting dictionaries to data frames

```
df=pd.DataFrame(launch_dict)
```

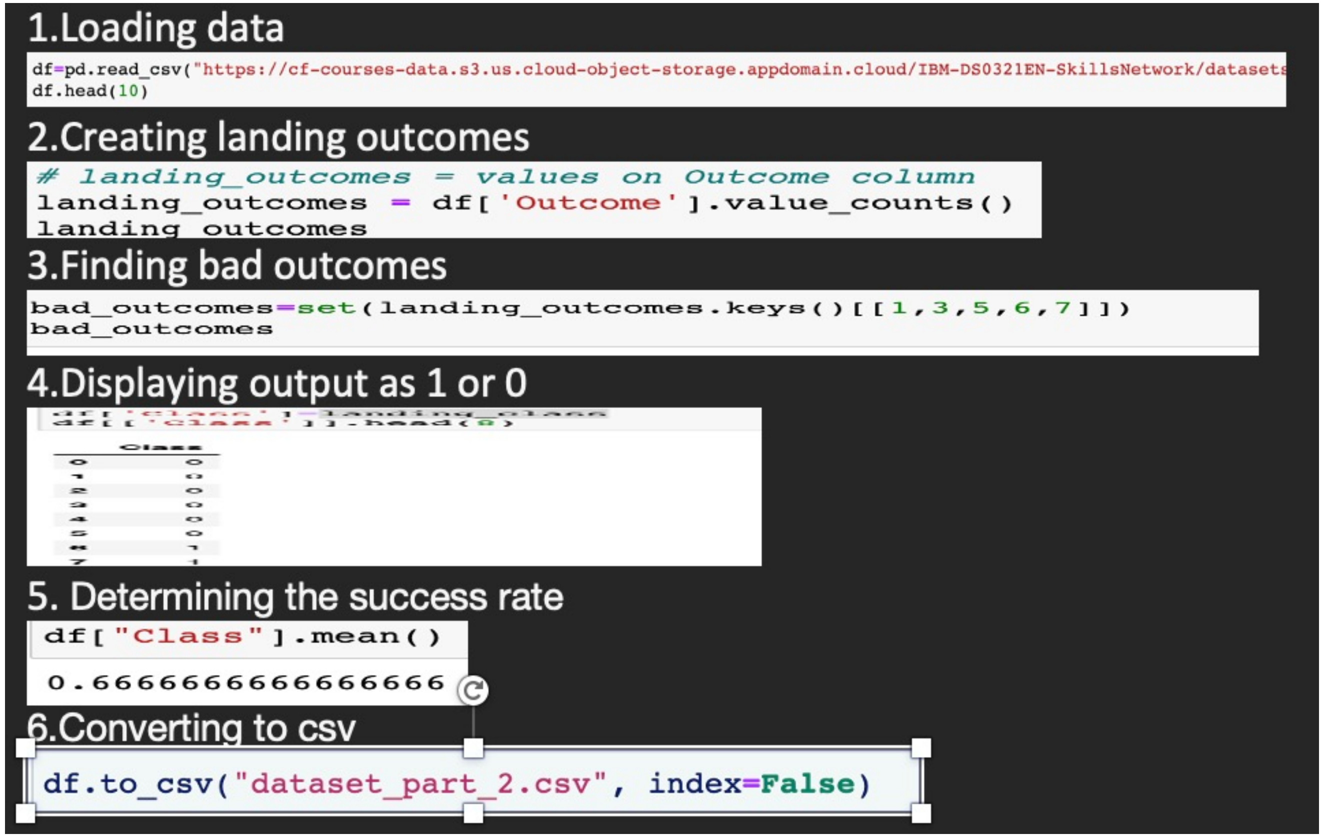
8.Converting data frames to csv file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

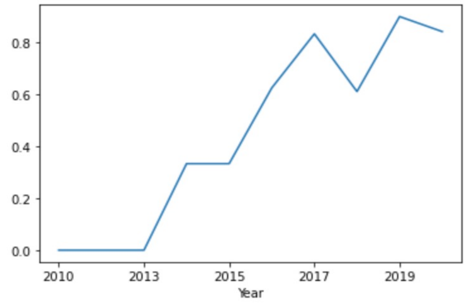
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

- [https://github.com/mingluzeng/spaceX_assignment/blob/main/labs-jupyter-spacex-Data wrangling.ipynb](https://github.com/mingluzeng/spaceX_assignment/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb)

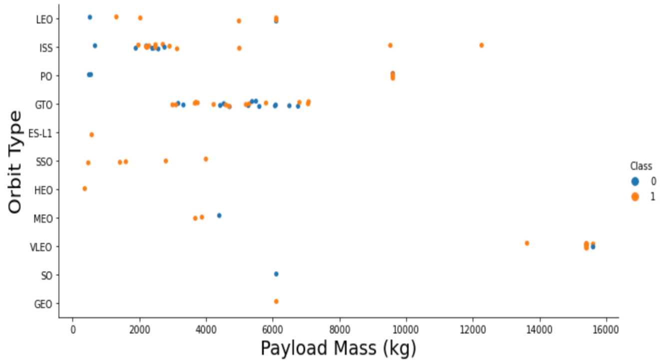


EDA with Data Visualization

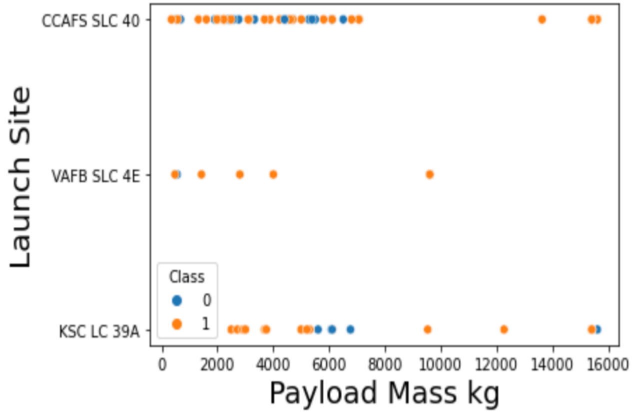
https://github.com/mingluzeng/spaceX_assignment/blob/main/jupyter-labs-eda-dataviz.ipynb



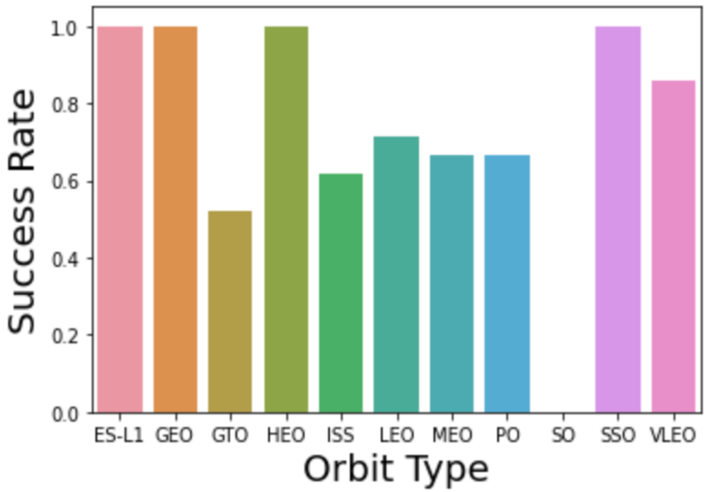
launch success yearly trend



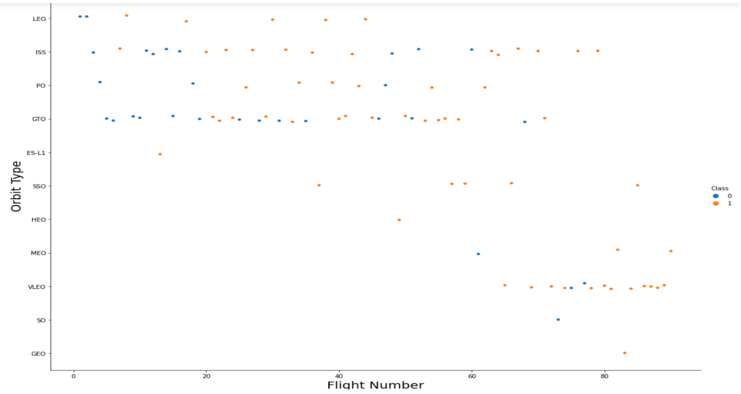
relationship between Payload and Orbit type



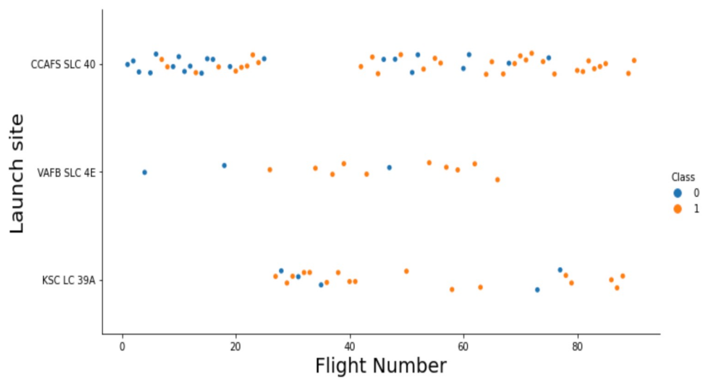
relationship between Payload and Launch Site



relationship between success rate of each orbit type



relationship between FlightNumber and Orbit type



relationship between Flight Number and Launch Site

EDA with SQL

Using bullet point format, summarize the SQL queries you performed

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

https://github.com/mingluzeng/spaceX_assignment/blob/main/Spacex (1).ipynb

Build an Interactive Map with Folium

Mark the success/failed launches for each site on the map to find an optimal location

https://github.com/mingluzeng/spaceX_assignment/blob/main/dashboard (1).ipynb

Build a Dashboard with Plotly Dash

We displayed the total launches by a certain sites by pie chart

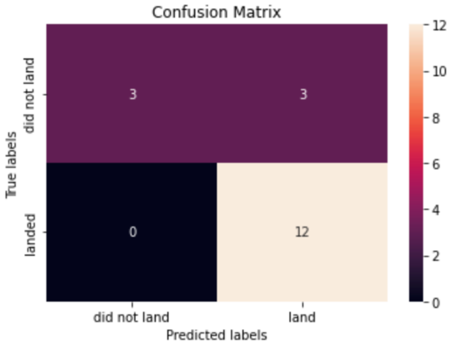
We visualized the relationship with Outcome and Payload by plotted scatter graph

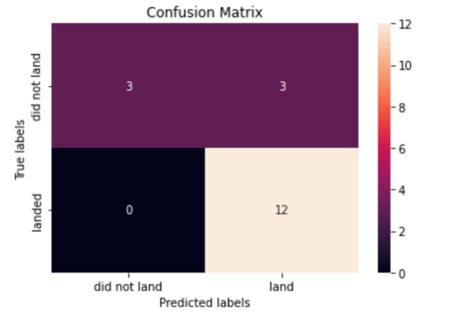
https://github.com/mingluzeng/spaceX_assignment/blob/main/Ploty_Dash.py

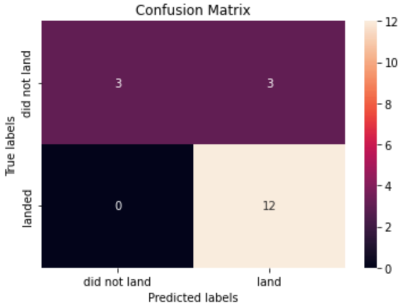
Predictive Analysis (Classification)

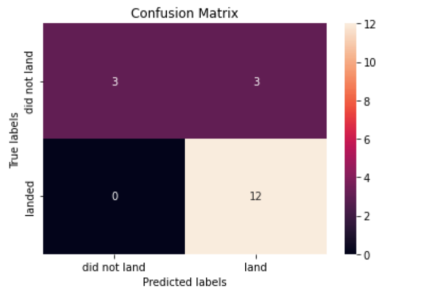
Build the models → Evaluate the models → Find the best model

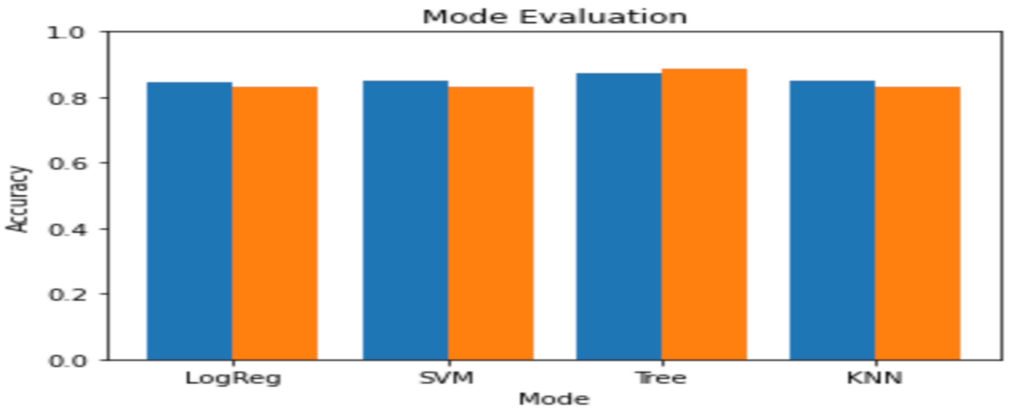
https://github.com/mingluzeng/spaceX_assignment/blob/main/SpaceX_Machine Learning Prediction_Part_5.ipynb











Decision tree has the highest score

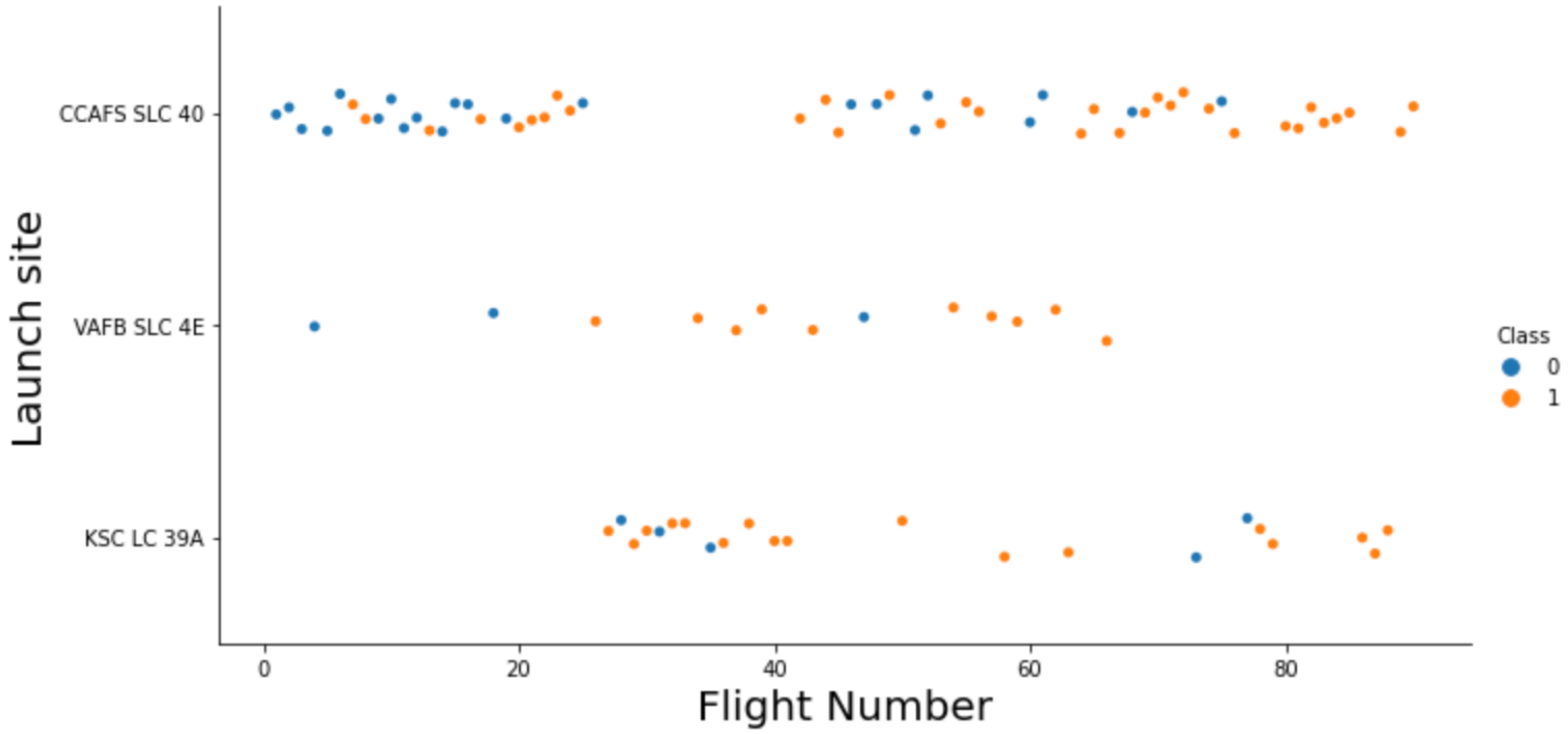
Results

Section 2

Insights drawn from EDA

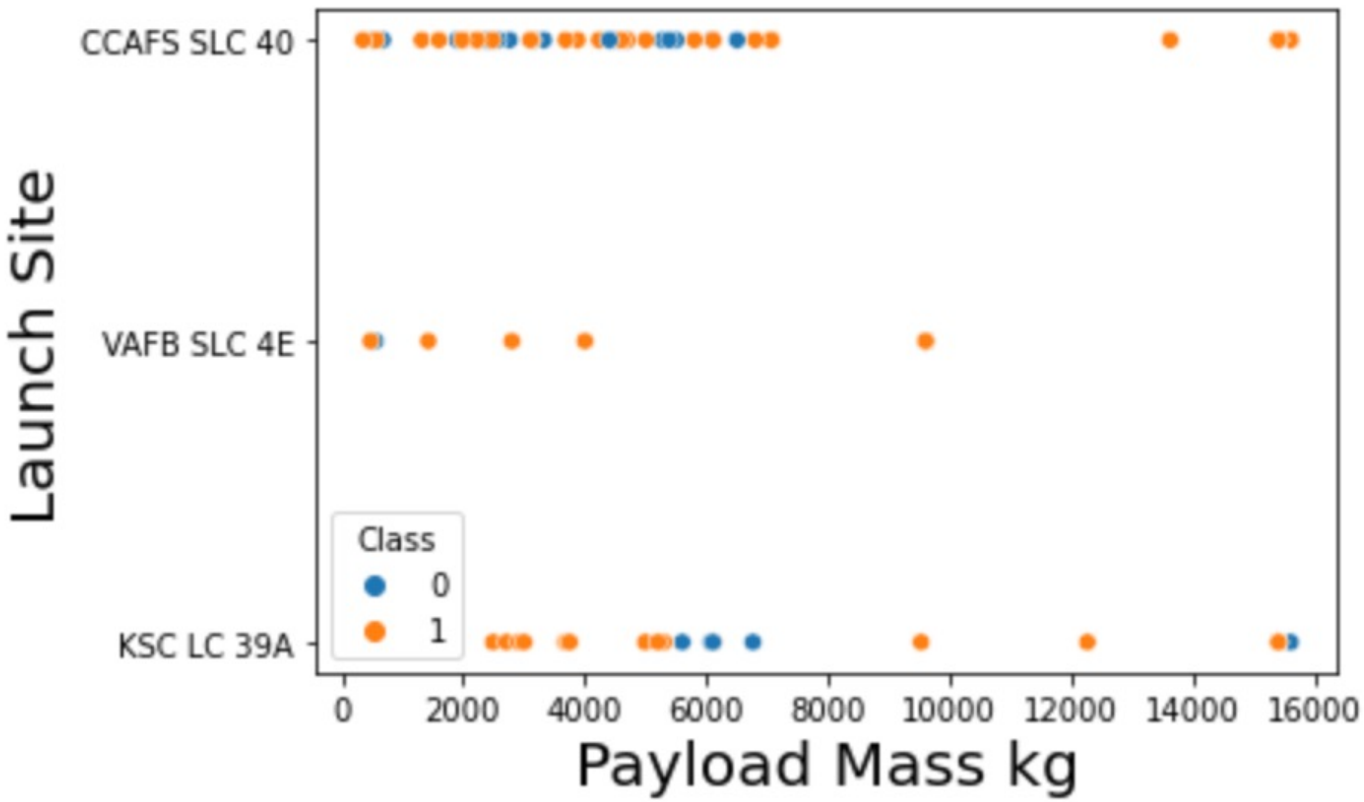
Flight Number vs. Launch Site

- Launch site of CCAFS SLC 40 is significantly higher than others



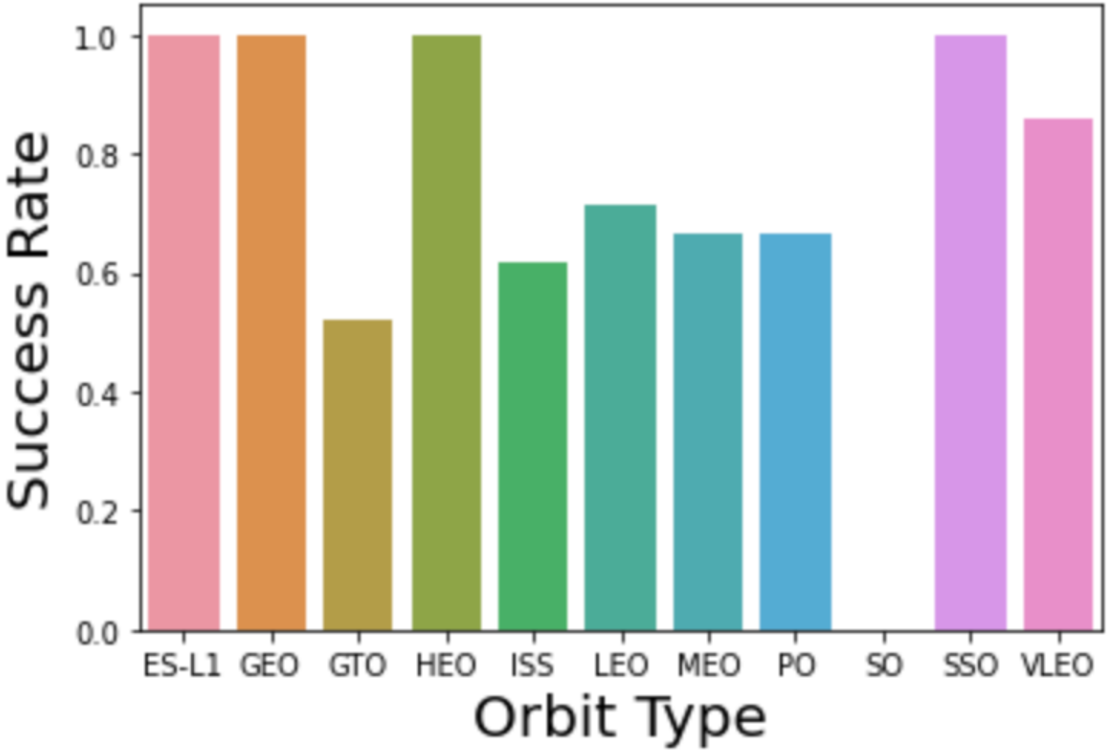
Payload vs. Launch Site

- The Majority of loads with lower mass are launched from CCAFS SLC 40



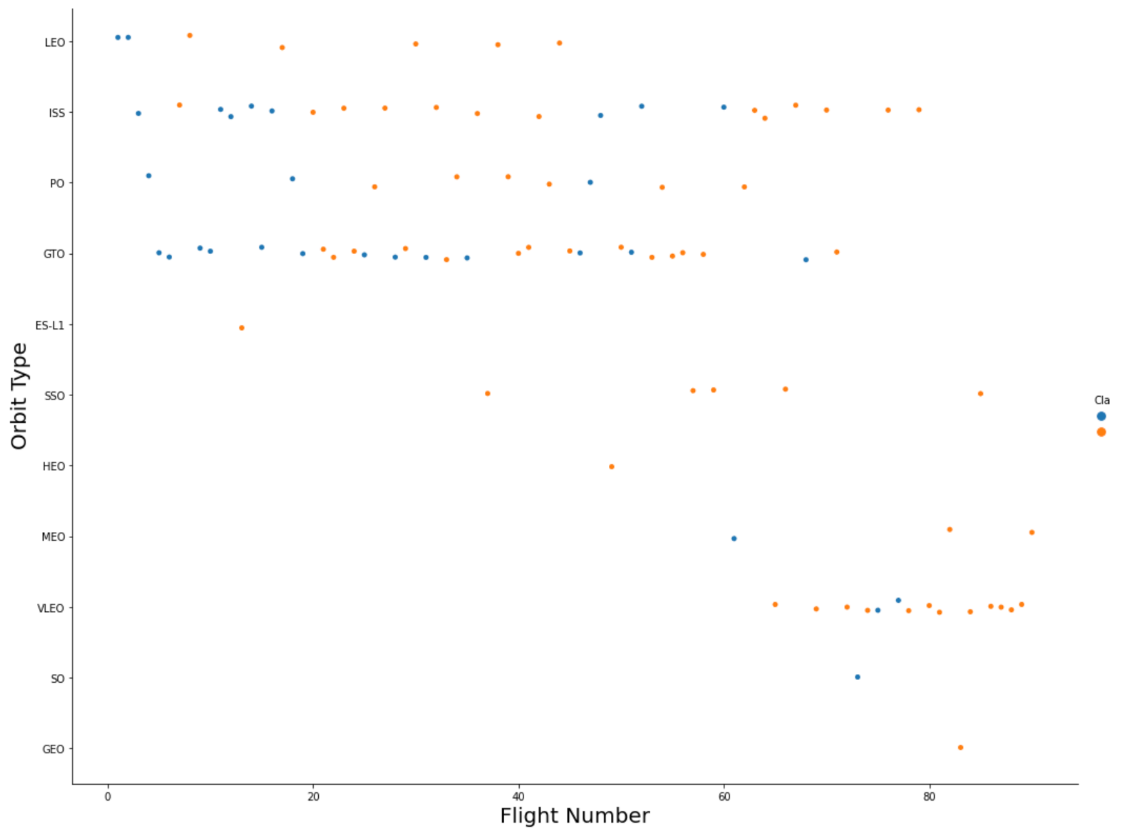
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO have the highest success rate



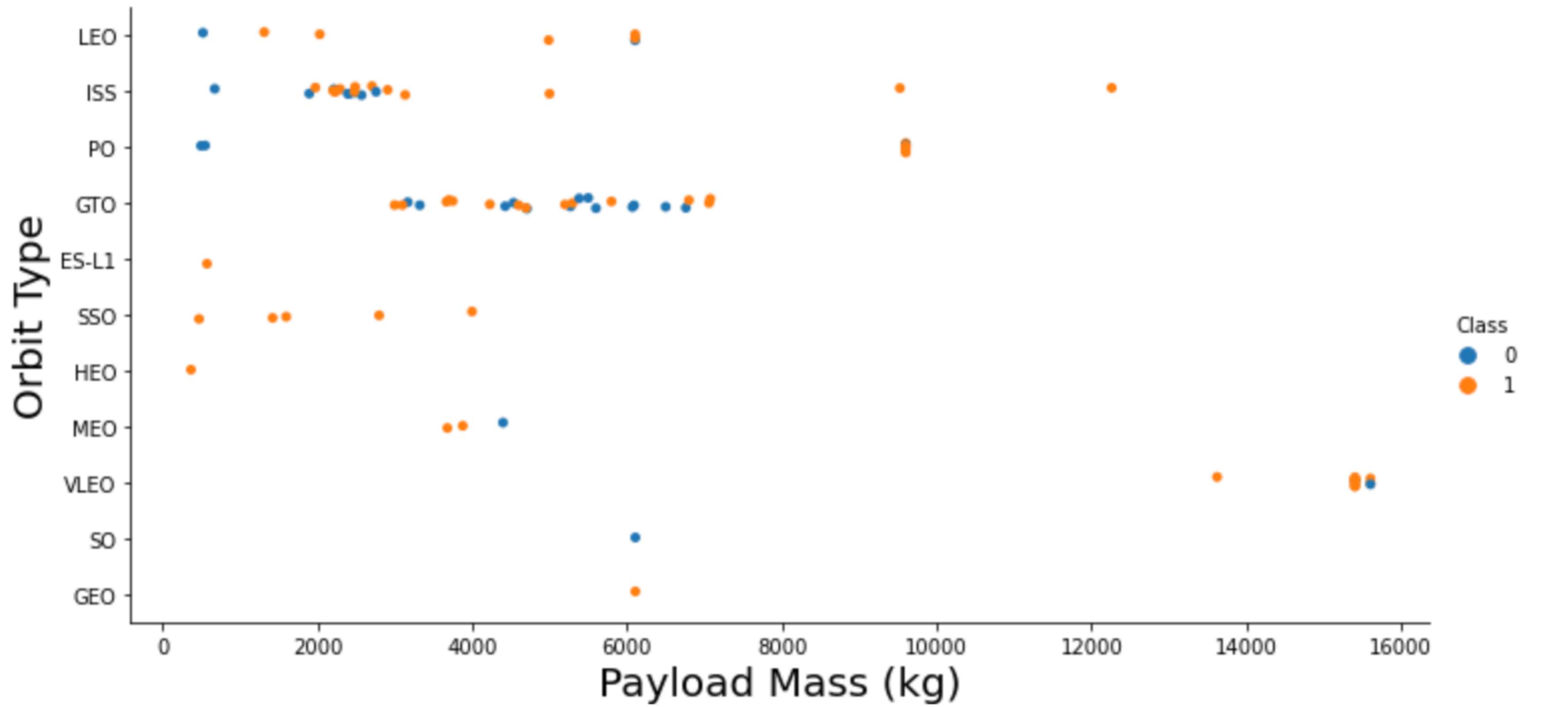
Flight Number vs. Orbit Type

- We can see the trend of shifting to VLEO in recent years



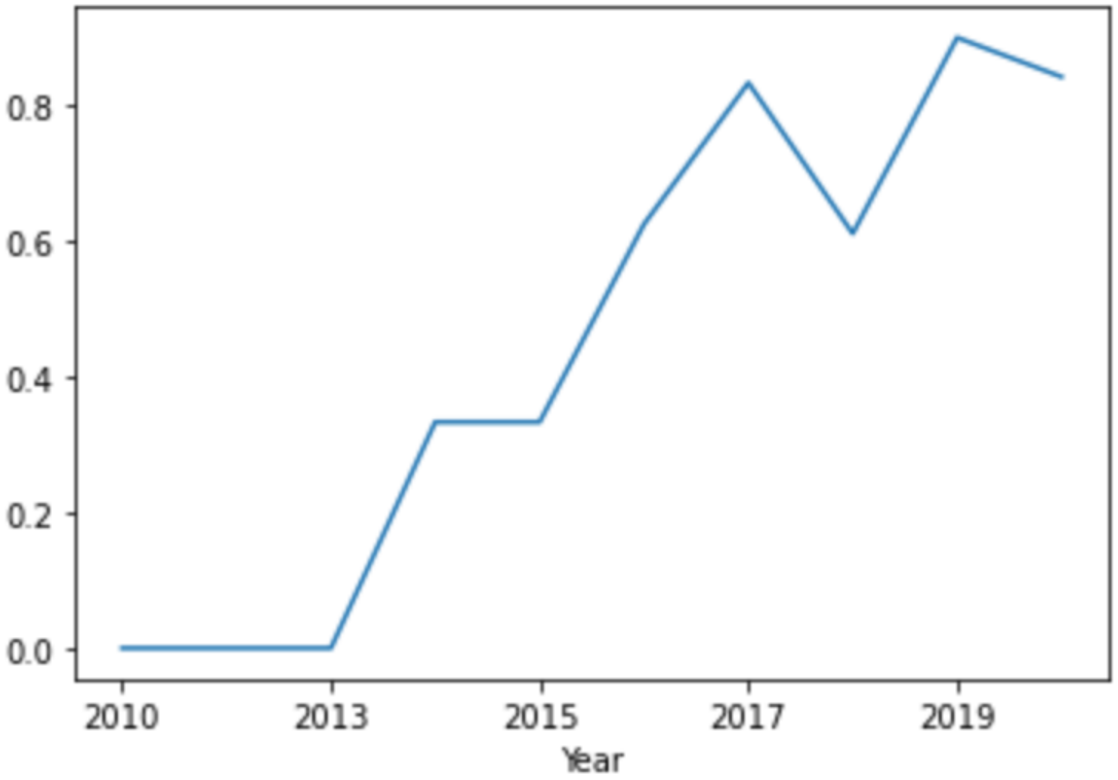
Payload vs. Orbit Type

- We can observe the strong correlation between ISS and Playload from 2000 to 3000, and GTO from 3000 to 8000



Launch Success Yearly Trend

you can observe that the Success rate since 2013 kept increasing till 2020



All Launch Site Names

- **Display the names of the unique launch sites in the space mission**

CCAFS LC-40
CCAFS SLC-40
KSC LC-39A

Launch Site Names Begin with 'CCA'

- **Display 5 records where launch sites begin with the string 'CCA'**

payload	payload_mass__kg_	orbit	customer	missi
ualification Unit	0	LEO	SpaceX	
ght C1, two of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	
aceX CRS-1	500	LEO (ISS)	NASA (CRS)	
ceX CRS-2	677	LEO (ISS)	NASA (CRS)	
SES-8	3170	GTO	SES	

Total Payload Mass

- **Display the total payload mass carried by boosters launched by NASA (CRS)**

Average Payload Mass by F9 v1.1

- **Display average payload mass carried by booster version F9 v1.1**

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://phy99914:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.dat
Done.
```

1
3676

First Successful Ground Landing Date

- **List the date when the first successful landing outcome in ground pad was acheived.**

```
%sql select min(DATE) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://phy99914:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.da
Done.
```

1
2017-01-05

Successful Drone Ship Landing with Payload between 4000 and 6000

- **List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000**
- In [10]:

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
]: %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success (drone s
```

```
 * ibm_db_sa://phy99914:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

]:

booster_version
F9 FT B1031.2
F9 FT B1022

Total Number of Successful and Failure Mission Outcomes

- **List the total number of successful and failure mission outcomes**

]:

```
%sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://phy99914:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.ap
Done.
```

]:

mission_outcome	2
Success	44
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- **The names of the booster_versions which have carried the maximum payload mass.**

```
%sql SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://phy99914:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600

2015 Launch Records

- **The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015**

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME='Failure (drone ship)' AND DATE LIKE '2015%'
```

```
 * ibm_db_sa://phy99914:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- **Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order**

```
%sql SELECT LANDING__OUTCOME, COUNT(*) AS qty FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME OI
```

```
 * ibm_db_sa://phy99914:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

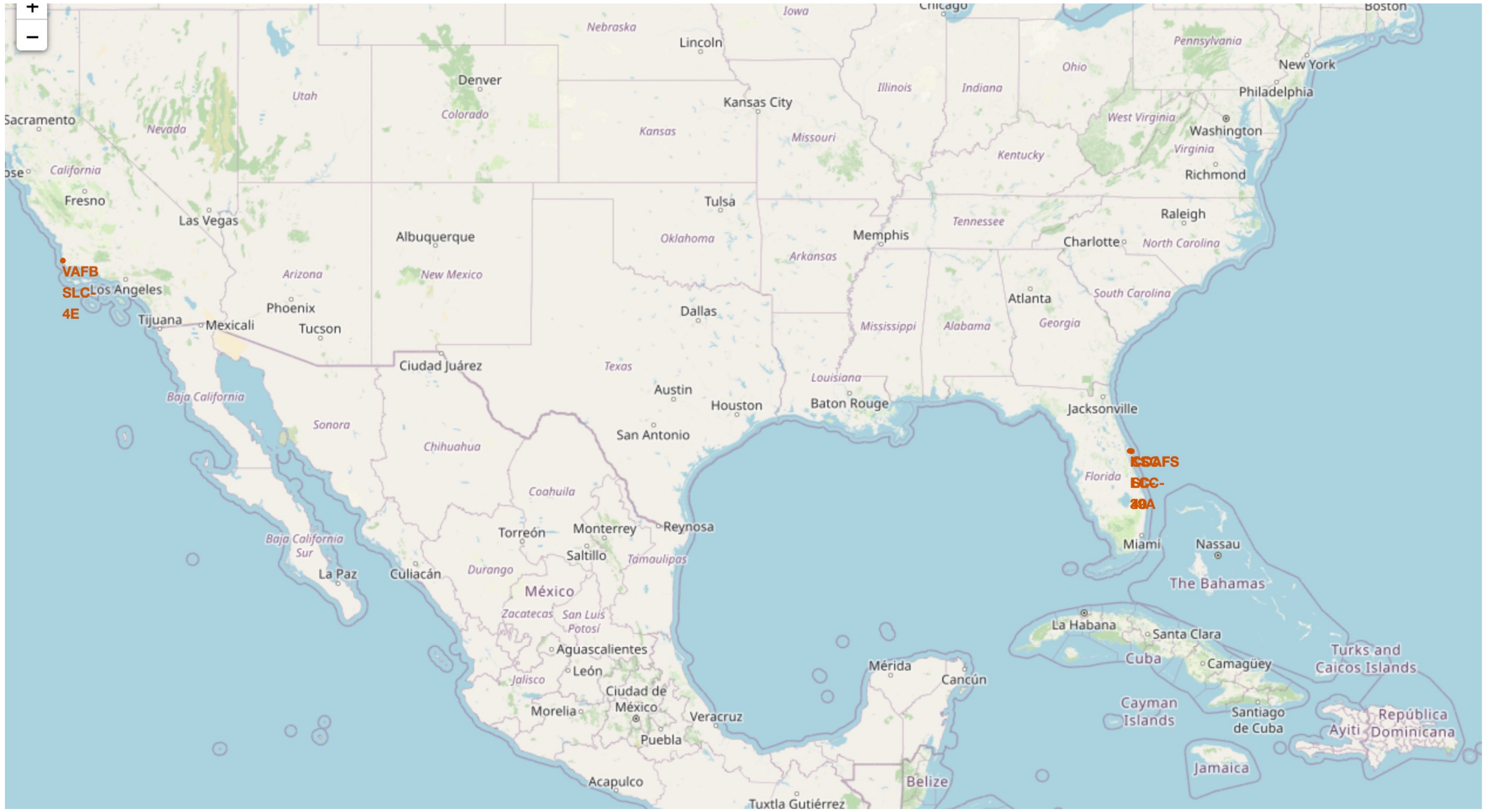
landing__outcome	qty
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- **all launch sites marked on a map**

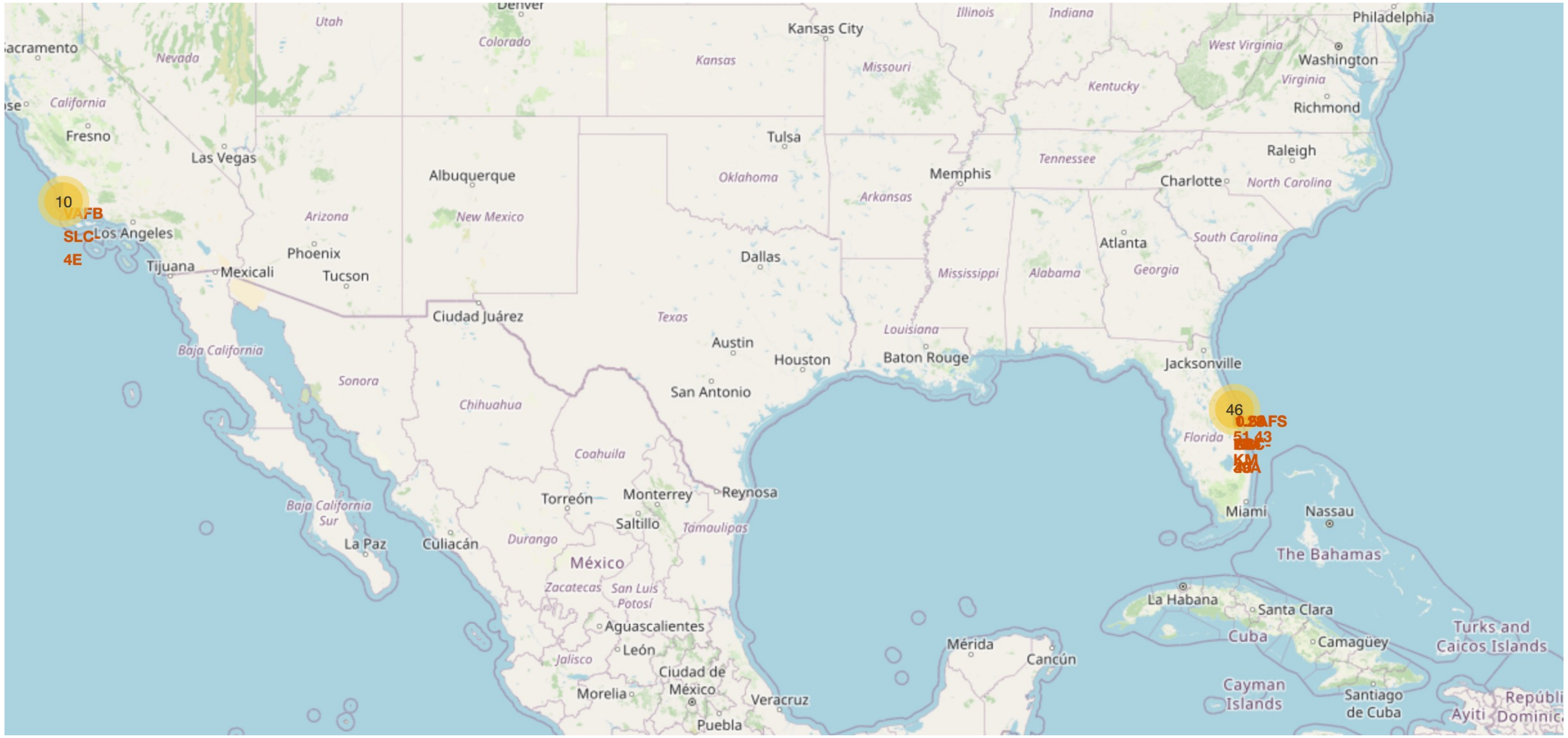


<Folium Map Screenshot 2>

- **The success/failed launches marked for each site on the map**

<Folium Map Screenshot 3>

- **The distances between a launch site to its proximities**

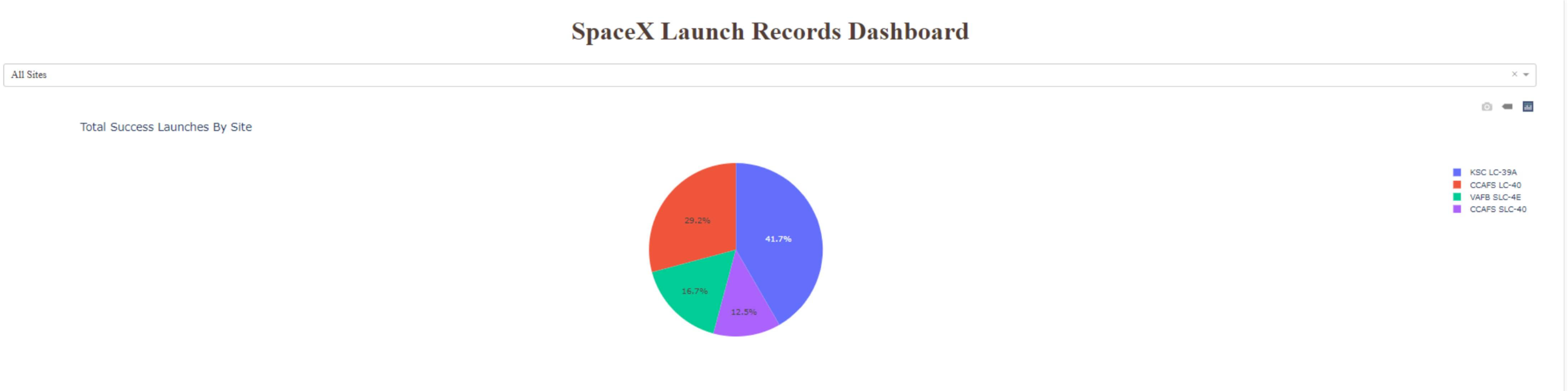


Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

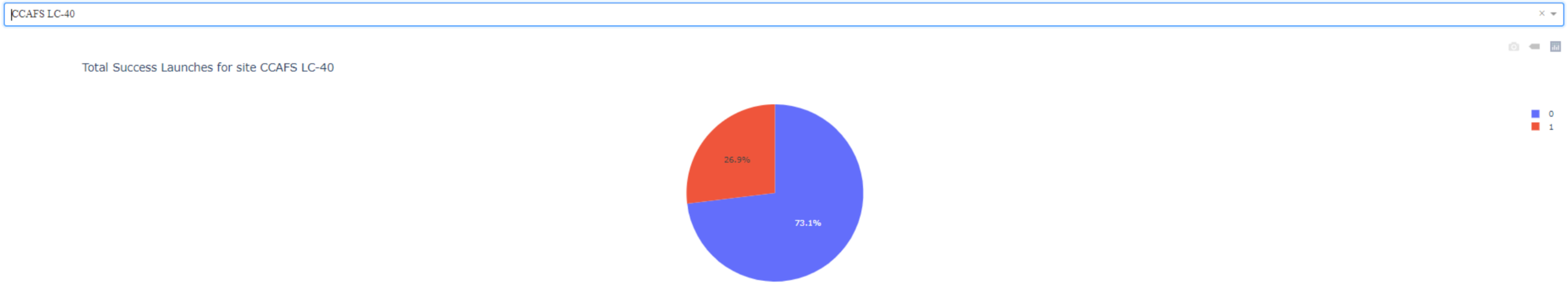
Total success launches by all sites



<Dashboard Screenshot 2>

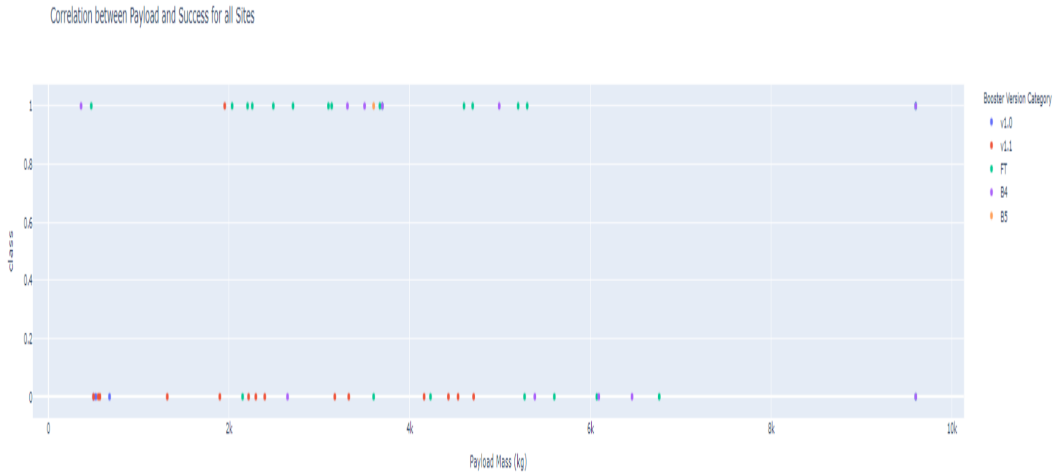
Success rate by site

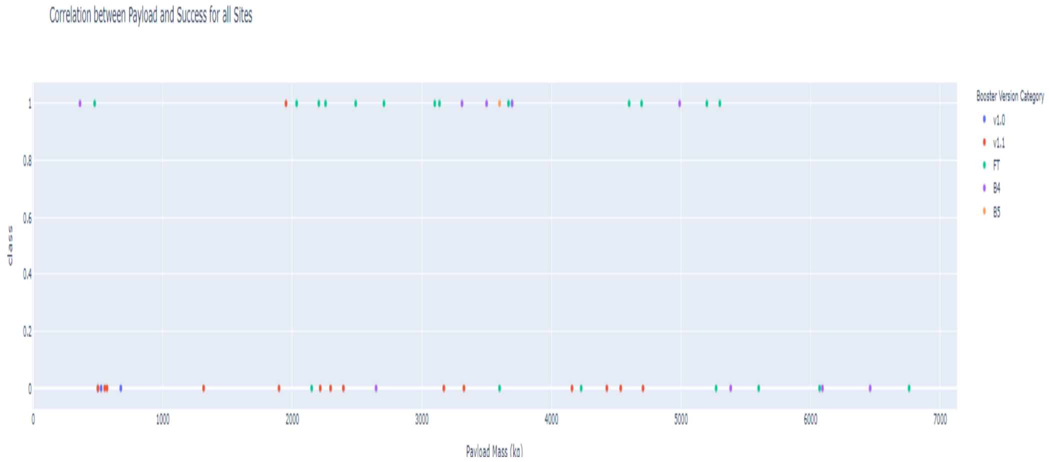
- Pie chart for is selected



<Dashboard Screenshot 3>

Payload vs launch outcome



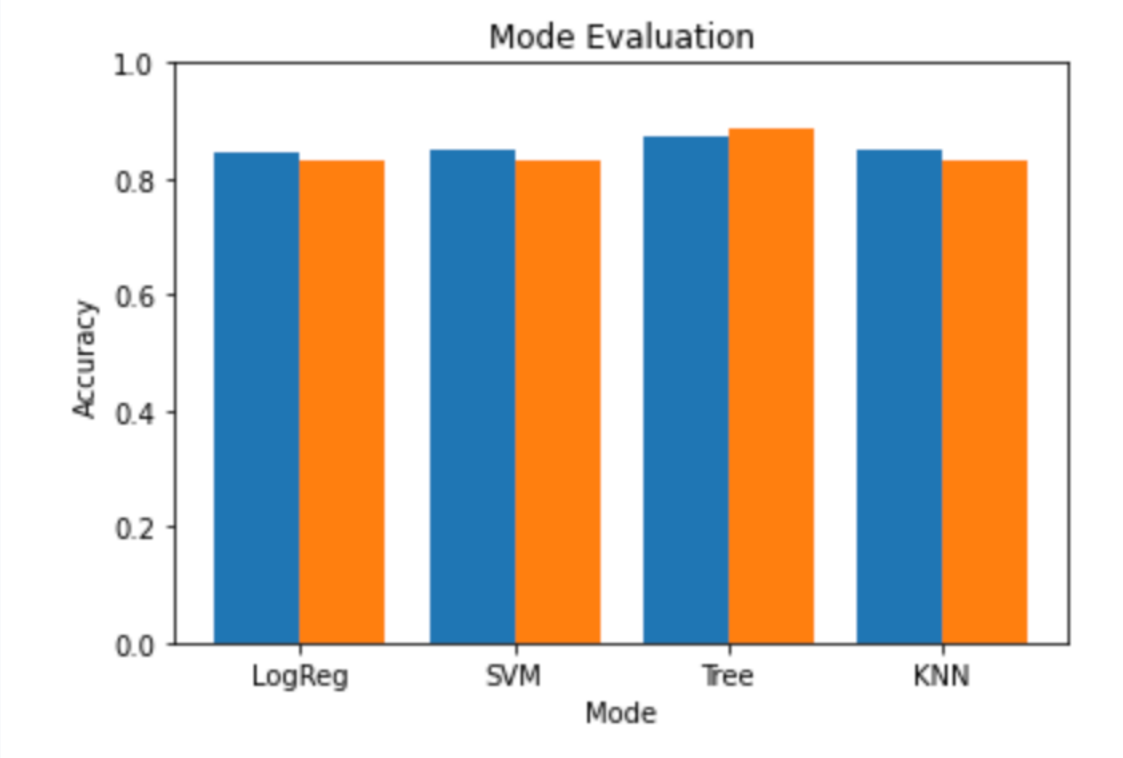


Section 5

Predictive Analysis (Classification)

Classification Accuracy

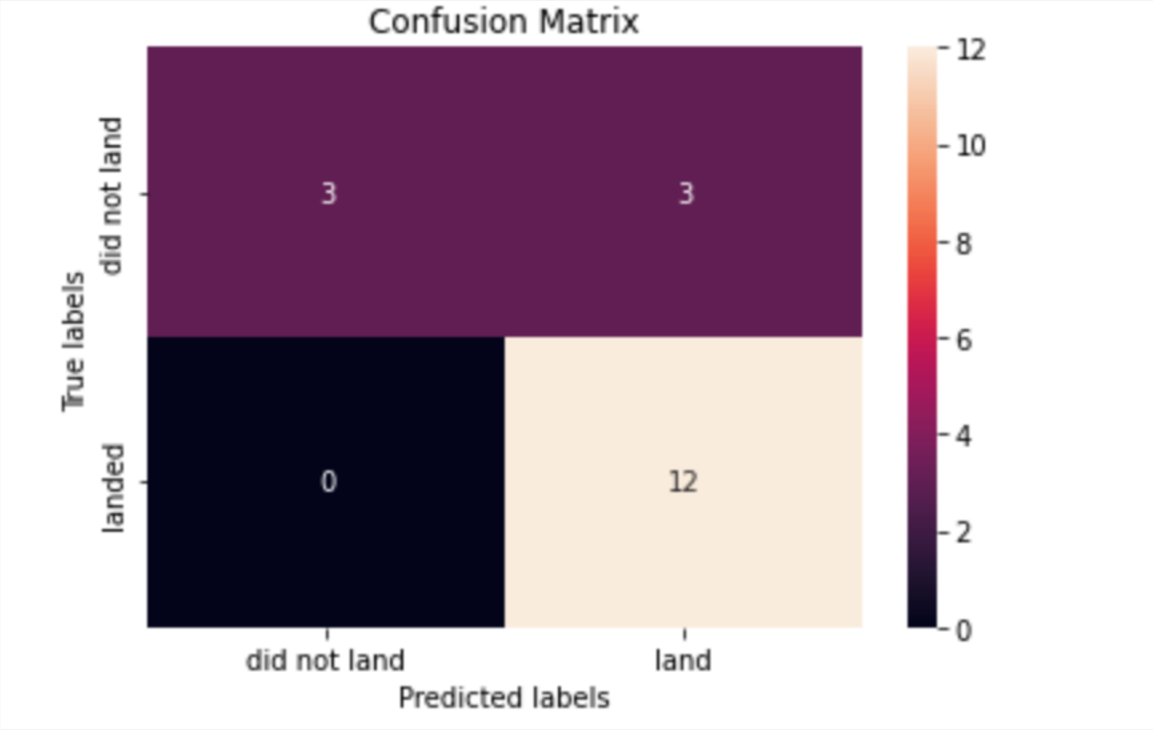
- Decision tree mode has the highest classification accuracy



Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.875	0.88889
KNN	0.84821	0.83333

Confusion Matrix

- It is the Confusion matrix of Decision Tree. It proves accuracy by showing the big numbers of true positive and true negative compared to the false ones.



Conclusions

- Decision tree mode has the highest accuracy
- Most landings are successful
- Launch success rate increased significantly in 2013 till 2020
- ES-L1, GEO, HEO, SSO have the highest success rate

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!